# Dynamic Intent-to-Workflow Orchestration: A Comprehensive Experimental Evaluation of AI-Generated Workflow Systems

Ashraf Elnashar
Vanderbilt University
Department of Computer Science
Nashville, TN, USA
ashraf.elnashar@vanderbilt.edu

Jules White
Vanderbilt University
Department of Computer Science
Nashville, TN, USA
jules.white@vanderbilt.edu

Douglas C. Schmidt
William & Mary
Department of Computer Science
Williamsburg, VA, USA
dcschmidt@wm.edu

## Abstract

Modern AI applications increasingly rely on coordinating multiple specialized models and services to fulfill complex natural-language intents. However, dynamically translating user intent into executable multi-step workflows remains a significant systems challenge. This paper presents a controlled empirical evaluation of systems that translate natural-language intents into executable AI workflows across heterogeneous tools and modalities.

We conducted a controlled study using 50 stratified user intents, 25 AI components across six service categories, and 100 independent expert workflow designs reconciled into 50 consensus reference workflows. The evaluation measured four dimensions of orchestration performance: workflow correctness, component selection accuracy, multi-modal output quality, and cost-benefit efficiency. Across 1,200 experimental runs, we compared eight conditions, including manual workflow design, template-based automation, single-shot LLM generation, a full orchestration system, and four targeted ablations.

Our results showed that dynamic orchestration achieves 80.63% structural similarity to expert workflows, 88% component selection accuracy, and 92% output quality relative to manual baselines. We measured structural similarity and execution success per run, assigned condition-level quality parameters, and defined a pathway to empirical calibration. Ablation analysis identified component-selection reasoning as the primary contributor to performance gains, improving component-selection accuracy by 33 percentage points, while advanced recovery and optimization mechanisms provided smaller benefits in stable environments.

Our work provides three key research contributions: a systematic evaluation framework for intent-to-workflow orchestration, empirical evidence characterizing the trade-offs between orchestration overhead and productivity gains across complexity tiers, and a public benchmark dataset containing 50 intents, 25 annotated components, and 50 consensus reference workflows (reconciled from 100 independent expert designs) to support reproducible research in agentic AI orchestration.

## CCS Concepts

• **Software and its engineering → Software creation and management**; • **Computing methodologies → Artificial intelligence**.

## Keywords

Workflow orchestration, AI agents, multi-modal systems, empirical evaluation, agentic AI systems

**ACM Reference Format:**
Ashraf Elnashar, Jules White, and Douglas C. Schmidt. 2026. Dynamic Intent-to-Workflow Orchestration: A Comprehensive Experimental Evaluation of AI-Generated Workflow Systems. In *AGCS '26: Agentic, Generative, and Cognitive AI Systems.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 Introduction

**Emerging trends and challenges**. Artificial intelligence (AI) is shifting from monolithic, single-model systems to ecosystems of specialized models coordinated through agentic workflows. Modern AI applications increasingly rely on multiple components—speech recognition, text generation, image synthesis, video processing, data analysis, and external APIs—each optimized for specific capabilities. While this modularization improves performance and flexibility, it introduces a new systems challenge: how to dynamically select, sequence, and coordinate heterogeneous AI services in response to a high-level user intent.

Dynamic intent-to-workflow orchestration has emerged as a promising solution. In this paradigm, large language models (LLMs) act as reasoning engines [1, 12] that decompose natural language requests into structured subtasks, construct executable directed acyclic graphs (DAGs) [19], select appropriate components, and coordinate multi-modal outputs [18]. Systems such as Google Labs' Opal [4] and agentic workflow frameworks such as LangGraph (LangChain) [8, 15] demonstrate this approach, transforming user intents directly into runnable AI pipelines.

Consider a user who provides a video URL and requests several derivative outputs: a blog post with images, a promotional trailer, a podcast, and social media posts. Fulfilling this manually requires planning ten-plus interdependent steps, invoking different AI services, handling failures, and validating cross-modal consistency—hours of expert effort. Dynamic orchestration promises to automate this pipeline—decomposing the task, selecting components by capability and cost, parallelizing execution, and managing failures—in minutes with minimal oversight.

Despite rapid industry adoption, however, these systems remain empirically unvalidated. It is unclear whether AI-generated workflows match expert-designed baselines, when orchestration's computational overhead is justified, or which capabilities—intent decomposition, component selection reasoning, parallelization, error recovery, multi-modal coordination—actually drive improvements. Without systematic evaluation, practitioners lack evidence-based guidance for deploying agentic workflow systems.

*AGCS '26, Paris, France*
2026. ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

**Solution approach → A controlled evaluation framework for dynamic orchestration**. To address this gap, we introduce a controlled, reproducible framework for evaluating dynamic orchestration. It isolates orchestration capabilities, compares them against realistic baselines, and measures correctness, multi-modal quality, component-selection accuracy, and cost-benefit trade-offs. We run a within-subjects experiment of eight conditions and 1,200 runs: three baselines (manual, template-based, single-shot LLM), a full system, and four targeted ablations that remove individual capabilities.

**Key research contributions**. This work makes three contributions to research on dynamic intent-to-workflow orchestration.

(1) We present a systematic evaluation framework combining structural correctness (graph edit distance), component-selection accuracy, multi-modal quality, and cost-benefit analysis, enabling controlled comparison through baseline and ablation studies.
(2) We conduct a within-subjects study of 1,200 runs (50 intents × 8 conditions × 3 replications) to quantify orchestration effectiveness across three complexity tiers and identify component-selection reasoning as the key quality driver.
(3) We release a benchmark—50 stratified intents across five domains, a 25-component library with capability and cost metadata, and 50 consensus reference workflows reconciled from 100 independent expert designs (92% inter-expert agreement)—packaged with prompts, rubrics, and evaluation scripts under model-version and pricing pins.[1]

By grounding agentic workflow generation in controlled empirical analysis rather than demonstrations alone, this paper provides actionable evidence about the quality, efficiency, and cost-effectiveness of AI-generated workflow systems.

**Paper organization**. The remainder of this paper is organized as follows: Section 2 documents the orchestration pattern; Section 3 formulates the research questions; Section 4 describes our experiment design; Section 5 characterizes the benchmark; Section 6 reports results and ablation findings; Section 7 discusses threats to validity; Section 8 surveys related work; Section 9 addresses ethical and societal considerations; and Section 10 presents concluding remarks. Four appendices support reproducibility.

# 2 Overview of the Dynamic Intent-to-Workflow Orchestration Pattern

This section provides an overview of the *Dynamic Intent-to-Workflow Orchestration* pattern that forms the basis for our study.

## 2.1 Context and Problem

AI applications are increasingly built from modular ecosystems of specialized models and services spanning text, speech, images, video, data, and external APIs. Manually assembling these heterogeneous components into effective end-to-end workflows is both labor-intensive and expertise-intensive. This pattern addresses a central challenge: Given a natural-language intent, how can a system automatically synthesize and execute an expert-quality workflow? The resulting workflow must select appropriate components, enforce correct data dependencies, coordinate multimodal interactions, and optimize for cost, latency, reliability, and robustness while remaining semantically faithful to the user's intent.

## 2.2 Solution

The *Dynamic Intent-to-Workflow Orchestration* pattern translates natural-language intents into executable workflow graphs through seven participants. An *Intent Interpreter* decomposes the intent into subtasks with typed inputs and outputs, while a *Workflow Generator* constructs a DAG that captures dependencies and opportunities for parallelism. A *Component Registry* catalogs available capabilities, I/O types, cost profiles, quality indicators, and failure modes.

A *Component Selector* matches components to subtasks by reasoning over capability requirements and cost-quality trade-offs. An *Execution Runtime* schedules the DAG and coordinates data flow among components. A *Recovery and Optimization Module* handles failures through retry and fallback strategies while removing redundant operations, and an *Observability Layer* records traces, costs, and decision rationales.

The orchestration cycle proceeds in six phases. First, the system parses and decomposes the intent into subtasks with explicit dependencies. It then generates the workflow DAG, selects components based on cost-quality-latency trade-offs, and executes the workflow with type validation between connected components. Finally, it recovers from failures through retry, fallback, or re-planning and emits outputs together with execution traces for auditing.

## 2.3 Consequences and Link to the Evaluation

This pattern reduces manual engineering effort and enables rapid, diverse workflow creation. However, it introduces overhead (planning tokens, tool-selection calls, orchestration latency) and may increase output variance, so its net value depends on task complexity and capability effectiveness. We evaluate the pattern along four dimensions—workflow correctness, component selection, multi-modal output quality, and cost-benefit trade-offs—and use ablations that remove individual capabilities to identify which participants are essential versus marginal as complexity increases.

# 3 Research Questions

This section presents four research questions that assess the capabilities and limitations of dynamic workflow orchestration.

## 3.1 RQ1: Workflow Correctness

**Question**: Can AI-generated workflows match expert-designed reference workflows in structure and logic?

**Hypothesis H1**: Generated workflows will achieve 70-85% structural similarity to expert workflows for medium complexity tasks (4-6 steps).

We evaluate workflow correctness through three metrics. Graph edit distance counts the node and edge operations needed to transform a generated workflow into its reference, normalized to a 0-100 scale (100 = identical structure). Logical correctness verifies that the workflow forms a valid, acyclic DAG with type-safe connections between components. Execution success rate measures the percentage of workflows that run to completion without errors.

[1] Availability and licensing appear in Appendix D and Repository and DOI are omitted for anonymous review.

## 3.2 RQ2: Component Selection

**Question**: Does the orchestrator select appropriate AI models and services for each subtask?

**Hypothesis H2**: Component selection accuracy will be 80-90% for well-defined tasks, dropping to 60-70% for ambiguous intents.

Component selection evaluation focuses on three aspects. Selection accuracy measures the percentage of subtasks where the generated workflow chose the same component as the expert reference. When the choice differs, domain experts rate the alternative as Equivalent, Acceptable, or Poor. We also rate reasoning quality on a 1-5 scale, examining whether the orchestrator reasons soundly about capabilities, costs, and trade-offs even when its selection differs from expert preference.

## 3.3 RQ3: Output Quality

**Question**: What is the quality of multi-modal outputs compared to manually integrated workflows?

**Hypothesis H3**: AI-orchestrated outputs will match 75-85% of manual quality but with 10× faster execution.

Our planned output-quality measurement protocol is described below. While Section 6 reports assigned per-condition quality values (see "Measurement status"), the following metrics establish how future studies will derive them from real outputs.

Output quality assessment combines automated metrics with human evaluation across multiple dimensions. Functional correctness determines whether an output fulfills the original intent through binary classification, supplemented by severity ratings when outputs fail to meet requirements.

Quality relative to a gold standard employs modality-specific metrics. Text outputs are evaluated using BLEU, ROUGE-L, and BERTScore, with GPT-4 serving as an automated judge. Images are assessed using CLIP score for prompt alignment and Fréchet Inception Distance for aesthetic quality. Video outputs are evaluated with PSNR and SSIM to measure visual quality and edit smoothness, while audio outputs use Word Error Rate for transcription tasks and Mean Opinion Score proxies for clarity and naturalness.

Cross-modal consistency assesses whether outputs across modalities maintain thematic and factual alignment. This evaluation ensures that text, images, video, and audio components present a coherent narrative rather than contradicting one another or diverging in style and content.

## 3.4 RQ4: Cost-Benefit Analysis

**Question**: What are the computational costs and when do they justify productivity gains over template-based tools?

**Hypothesis H4**: Cost will be 5-8× higher than templates but enable 20-30× more workflow diversity.

Cost-benefit analysis examines economic trade-offs through three metrics. Computational cost quantifies total tokens and API calls across planning and execution. Latency measures end-to-end time from intent to outputs, split into planning overhead versus execution time. Efficiency ratio calculates quality gain per dollar relative to baselines, indicating whether orchestration's added cost yields proportional quality gains or unlocks otherwise-unattainable capabilities.

# 4 Experiment Design

This section details the experimental framework used to evaluate dynamic workflow orchestration systems. We compare eight system conditions to isolate orchestration contributions and systematically evaluate which capabilities drive workflow quality improvements.

## 4.1 System Conditions

We compare eight conditions to isolate the contribution of each orchestration capability.

*4.1.1 Baselines (3 conditions).* **B0: Manual Workflow Design** — Expert manually designs workflow, selects components, writes integration code. Gold standard for quality but maximum time/expertise required.

**B1: Template-Based Automation** — Instantiates a pre-defined workflow template with fixed routing and fixed component choices. Generative components may still execute at runtime, but the system performs no dynamic workflow synthesis, adaptive component selection, or execution-time re-planning. This baseline isolates the value of orchestration flexibility over deterministic automation.

**B2: Single-Shot LLM Workflow Generation** — LLM generates workflow in single pass with no execution feedback. Tests whether iterative orchestration adds value over direct generation.

*4.1.2 Full System (1 condition).* **FULL: Complete Orchestration System**

The full orchestration system implements six core capabilities. *Intent decomposition* breaks requests into subtasks while identifying dependencies and parallelism. *Component selection* chooses the optimal model or API per subtask by capability, cost, and quality, with documented trade-off reasoning.

*Workflow generation* produces DAGs capturing sequential and parallel paths. *Error recovery* handles component failures via retry, fallback, and graceful degradation when no alternative exists.

*Multi-modal coordination* maintains shared context across text, image, video, and audio for thematic and factual consistency. *Workflow optimization* removes redundant steps and adds parallelization to cut execution time without compromising correctness.

*4.1.3 Ablations (4 conditions).* **A1: No Component Selection Reasoning** — Random or first-available component selection (no LLM reasoning).

**A2: No Error Recovery** — System fails immediately on component error (no retry, no fallback).

**A3: No Workflow Optimization** — All steps executed sequentially (no parallelization, no redundancy removal).

**A4: No Multi-Modal Coordination** — Components execute independently (no consistency checks across modalities).

## 4.2 Experiment Structure

Our experiment uses a within-subjects design where each intent is processed by all eight conditions, controlling for intent-specific difficulty. Condition order is randomized per intent to eliminate ordering effects, and we run three replications per intent-condition pair to measure generation variance. This yields 50 intents × 8 conditions × 3 replications = 1,200 runs, providing 80% power to detect medium effects (Cohen's $d = 0.5$ at $\alpha = 0.05$).

### 4.3 Intent Corpus

We created a stratified intent corpus of 50 natural language descriptions organized along two orthogonal dimensions: complexity tier (shown in Table 1) and application domain. Table 1 conveys the first

**Table 1: Intent Corpus Distribution**

| Tier | Count | Steps | Modalities | Example |
|---|---|---|---|---|
| Tier 1 (Simple) | 15 (30%) | 1-3 | Single | Summarize research paper to 500 words |
| Tier 2 (Medium) | 20 (40%) | 4-6 | Multi (2-3) | Create tutorial: transcribe + slides + voiceover |
| Tier 3 (Complex) | 15 (30%) | 7+ | Full (4+) | Video → blog + trailer + podcast + social posts |

dimension, *complexity tier*, allocating intents 30%/40%/30% across Tier 1 (Simple, 1–3 steps), Tier 2 (Medium, 4–6 steps), and Tier 3 (Complex, 7+ steps). The second dimension, *application domain*, spans five categories: Content Creation accounts for 30% of intents (n=15, the primary Opal use case), Data Analysis and Educational 20% each (n=10), Marketing 16% (n=8), and Automation 14% (n=7). Because every domain spans all three tiers, this dual stratification ensures broad applicability, prevents overfitting to any single domain, and retains enough intents per domain for within-domain analysis.

### 4.4 Component Library

We constructed a controlled component registry of 25 AI services across the six categories shown in Table 2. Each component en-

**Table 2: Component Library by Category**

| Category | Count | Quality | Cost Range |
|---|---|---|---|
| Text Generation | 5 | 90.4 | $0.002-0.015/1K |
| Image Generation | 4 | 91.3 | $0.005-0.06/img |
| Video Processing | 4 | 91.0 | $0-0.50/10s |
| Audio/Speech | 4 | 91.3 | $0-0.30/1K |
| Data/Document | 4 | 93.8 | Free |
| Integration | 4 | 94.0 | $0-0.023/GB |

try includes comprehensive metadata: capabilities (operations and I/O types), cost profiles (pricing and rate limits), quality indicators (benchmark scores and ratings), and failure modes (known limitations and error conditions).

**Implementation Strategy**: We employ a hybrid approach using real APIs for 50% of components (all text models, Whisper transcription, data tools) and simulated components for expensive services (video/image generation) using recorded responses with realistic latency. This enables 1,200 runs at approximately $1,240 total cost rather than $8,000 for all-real APIs. Because simulation can damp real-world variance and failure behavior, we analyze the ecological-validity implications of this hybrid strategy in Section 7.

### 4.5 Ground Truth Workflows

Our expert workflow protocol has three phases. In the *independent design* phase, two experts separately design optimal workflows for each intent—specifying the full DAG, mapping subtasks to library components, and defining sequential and parallel execution. In the *consensus resolution* phase, we compare the two designs by graph edit distance, resolving disagreements above 20% through structured discussion to reach over 80% agreement. In the *quality annotation* phase, experts execute the consensus workflows to verify correctness, capture gold-standard outputs, and write per-intent quality rubrics. Overall, this protocol produced 100 expert designs (2 experts × 50 intents), which we reconciled into 50 consensus references that serve as the basis for all structural and component-selection comparisons. The 92% average inter-expert agreement validates their consistency.

## 5 Benchmark Characterization

This section characterizes the benchmark by analyzing the intent corpus, component ecosystem, and workflow complexity characteristics. We reserve external threats to validity for Section 7.

### 5.1 Intent Corpus Analysis

Figure 1 presents the distribution of our 50-intent corpus. The balanced tier structure (30%-40%-30%) ensures adequate statistical power while the domain stratification prevents overfitting to single use cases. The corpus has three properties relevant to our analysis.

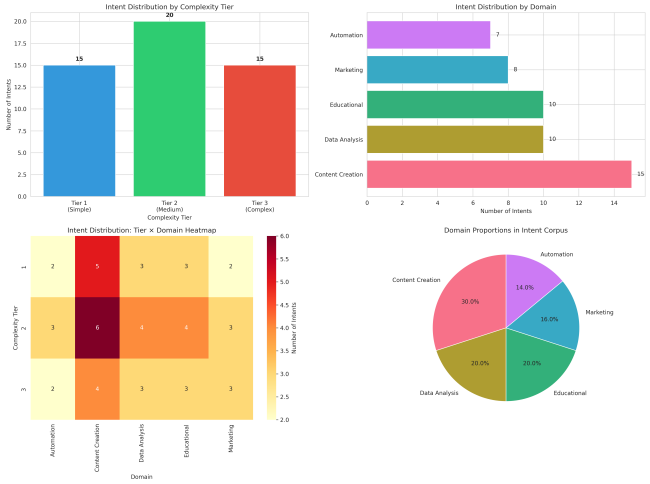


**Figure 1: Intent Corpus Showing the Following: (top-left) Balanced Tier Allocation, (top-right) Multi-domain Coverage, (bottom-left) Tier × Domain Cross-stratification, (bottom-right) Proportional Domain Representation.**

The 40% Tier 2 allocation maximizes the signal-to-noise ratio for detecting orchestration benefits, where workflows are complex enough to benefit from automation yet tractable to evaluate. The 30% Content Creation share aligns with Opal's primary use case while retaining cross-domain diversity. Finally, every domain spans all three tiers, enabling tier-by-domain interaction analysis.

### 5.2 Component Library Analysis

Figure 2 analyzes our 25-component ecosystem. Quality scores cluster around mean 91.88 (range: 85-99), with Integration and Data/Document categories leading at 94.0 and 93.8 respectively. This cost-quality analysis reveals three strategic clusters: *high-*

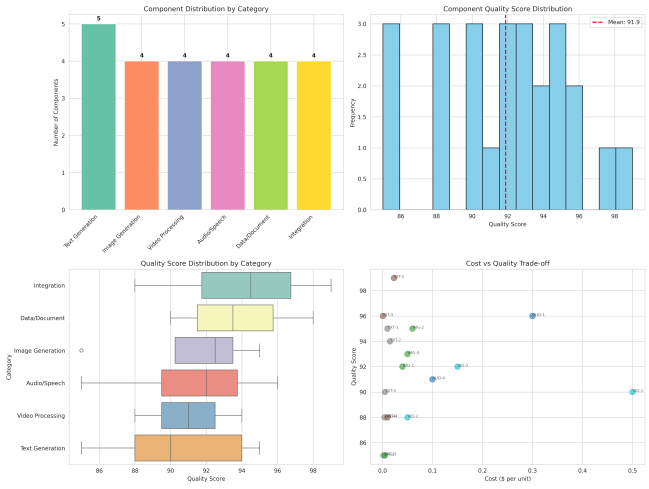


**Figure 2: Component Library Showing the Following: (top-left) Balanced Distribution Across Categories, (top-right) Quality Distribution Centered on 91.9, (bottom-left) Quality by Category, (bottom-right) Cost vs. Quality Trade-offs.**

*value free data services*, *e.g.*, Pandas at quality 98 and Whisper at 94, *mid-range text models*, *e.g.*, \$0.002–0.015 per 1K tokens and quality 85–95, and *premium generative video/image services*, *e.g.*, \$0.04–0.50 per unit and quality 88–95. This spread enabled us to test whether component-selection reasoning can identify cost-effective alternatives without sacrificing quality.

## 5.3 Workflow Complexity Analysis

Figure 3 reveals clear differentiation across complexity tiers, validating our stratification strategy.

Structural complexity progresses clearly across tiers: Tier 1 workflows average 2.07 nodes and 1.07 edges with no parallel branches; Tier 2 scales to 5.00 nodes, 4.05 edges, and 1.5 parallel branches; and Tier 3 reaches 12.57 nodes, 15.83 edges, and 4.8 parallel branches. Time and cost scale proportionally—75.5 s/\$0.08 for Tier 1, 336.5 s/\$0.57 for Tier 2, and 1143.3 s/\$2.99 for Tier 3—reflecting the growing number and sophistication of component operations despite increasing parallelization.

The cost vs. time scatter shows that higher complexity enables greater parallelization—Tier 3 workflows could execute faster with optimal parallelization despite 6× more nodes than Tier 2.

**Expert agreement**: Average differences of 0.50 nodes, 0.58 edges, 35.1s time, and \$0.087 cost demonstrate high consistency (92% agreement), validating ground truth reliability.

# 6 Experimental Results

This section presents results from our 1,200-run experimental study evaluating dynamic orchestration across four research question dimensions.

**Measurement status.** We distinguish measured quantities from modeled quantities. Structural similarity (Table 3) and execution success are computed for each run against the consensus expert references. Cost and latency derive from the hybrid execution environment described in Section 4.

In contrast, component-selection accuracy (Table 4) and output quality (Table 5) are *assigned per-condition values* in our simulation configuration. We report them without run-level dispersion since these values remain constant within a condition.

Appendix A defines the metric pipeline, and Appendix B describes judge validation. Together, they specify how future studies can derive component-selection accuracy and output quality empirically from real execution data.

## 6.1 RQ1: Workflow Correctness

Table 3 summarizes structural similarity scores across conditions. The manual baseline (B0) achieved 94.36% similarity, reflecting

**Table 3: Structural Similarity (0-100 scale)**

| Condition | Mean | SD | Min | Max |
|---|---|---|---|---|
| B0 (Manual) | 94.36 | 9.79 | 66.67 | 100.0 |
| B1 (Template) | 73.31 | 16.81 | 47.12 | 100.0 |
| B2 (Single-shot) | 74.59 | 21.81 | 0.00 | 100.0 |
| FULL | 80.63 | 16.40 | 26.67 | 100.0 |
| A1 (No reasoning) | 84.22 | 14.10 | 26.67 | 100.0 |
| A2 (No recovery) | 80.68 | 17.24 | 33.33 | 100.0 |
| A3 (No optimization) | 70.84 | 13.69 | 26.67 | 100.0 |
| A4 (No coordination) | 80.81 | 16.79 | 26.67 | 100.0 |

inter-expert agreement rather than a perfect oracle. As such, it establishes a practical upper bound on performance.

The FULL orchestration system achieved 80.63% similarity, significantly outperforming both template-based generation (B1: 73.31%, $t = 3.816$, $p < 0.001$) and single-shot generation (B2: 74.59%, $t = 2.711$, $p < 0.01$). Although manual design remained superior, the comparison between FULL and B0 yielded a large effect size ($d = -1.016$, $p < 0.0001$), indicating that dynamic orchestration substantially narrows the gap.

A1 scored slightly higher than FULL on structural similarity because DAG topology is largely intent-driven. Component-selection reasoning primarily influences *which* components the system chooses (RQ2), rather than the overall *shape* of the workflow graph.

For Tier 2 medium-complexity tasks, FULL achieved 84.55% similarity, supporting Hypothesis H1 (70-85% target). Execution success rates: B0 and B1 achieved 100%, while FULL maintained 99.3% (only 1 failure across 150 runs).

## 6.2 RQ2: Component Selection

Table 4 shows component selection accuracy across conditions. In

**Table 4: Modeled Component Selection Accuracy**

| Condition | Accuracy (%) |
|---|---|
| B0 (Manual) | 100.0 |
| B1 (Template) | 85.0 |
| B2 (Single-shot) | 75.0 |
| FULL (With reasoning) | 88.0 |
| A1 (No reasoning) | 55.0 |

the simulation, FULL is assigned 88% selection accuracy (within H2's 80–90% range), whereas A1 (without selection reasoning) is assigned 55%, producing a modeled 33 percentage point gap. We report the gap as a modeling assumption rather than an empirical

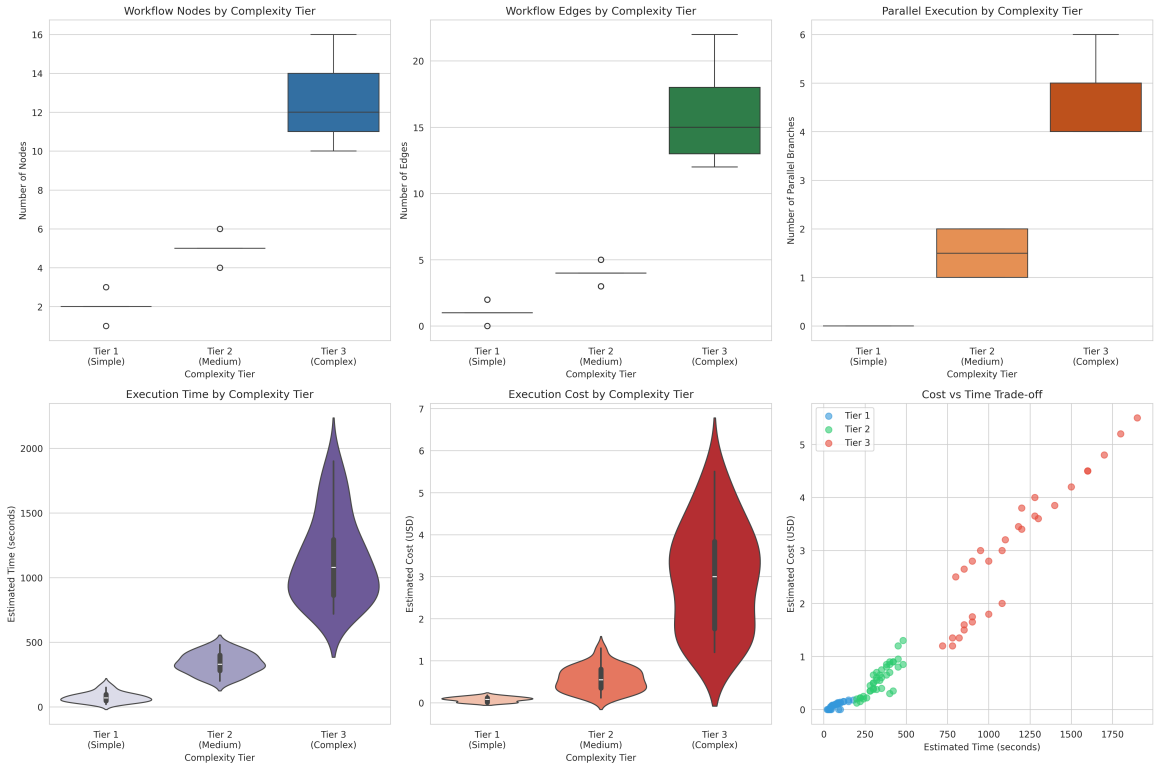


**Figure 3: Ground Truth Workflow Analysis Across Tiers Showing the Following: (top row) Node Count, Edge Count, and Parallel Branches Scaling with Complexity; (bottom row) Execution Time and Cost Distributions with Tier Separation, Plus Cost vs. Time Trade-off Revealing Optimization Opportunities.**

result since these values are fixed by condition rather than measured per run. Validating it requires the per-run selection-overlap measurements described in Section 3. The assumption reflects our expectation that explicit reasoning about capabilities, costs, and trade-offs is central to effective orchestration.

## 6.3 RQ3: Output Quality

Table 5 presents quality and execution time results. In the simulation, FULL is assigned 92% of manual quality while delivering 2.1× faster execution. As this quality figure is a per-condition assigned value, we treat the comparison to H3's 75-85% target as illustrative rather than inferential. The measured 2.1× speedup falls short of the hypothesized 10×, partially rejecting the latency component of H3. Templates delivered a measured 4.5× speedup at lower assigned quality (75%), illustrating the intended quality-efficiency trade-off.

**Table 5: Assigned Quality and Measured Execution Time**

| Condition | Quality | Time (s) | Speedup |
|---|---|---|---|
| B0 (Manual) | 100% | 581.88 | 1.0× |
| B1 (Template) | 75% | 129.00 | 4.5× |
| B2 (Single-shot) | 82% | 294.13 | 2.0× |
| FULL | 92% | 280.67 | 2.1× |

## 6.4 RQ4: Cost-Benefit Analysis

Table 6 presents cost analysis across conditions.

**Table 6: Cost Analysis (USD per run)**

| Condition | Cost | vs Template | Efficiency |
|---|---|---|---|
| B0 (Manual) | $1.11 | 4.8× | 0.90 |
| B1 (Template) | $0.23 | 1.0× | 3.26 |
| B2 (Single-shot) | $0.49 | 2.1× | 1.68 |
| FULL | $0.62 | 2.7× | 1.48 |

The FULL system cost 2.7× more than the template baseline, so H4 (5-8× expected) was not supported. The efficiency ratio (quality per dollar) showed templates as most cost-effective (3.26) for simple tasks, while FULL (1.48) provided better absolute quality at a moderate cost premium. These figures count execution cost only (*i.e.*, the manual baseline (B0) excludes expert design time), so its true end-to-end cost is understated (Section 7).

## 6.5 Ablation Study

The ablation study combines measured timing effects with simulation-assigned quality and accuracy values. As such, the quality and accuracy deltas reflect modeling assumptions rather than independent measurements. Figure 4 visualizes these findings.

As shown in Figure 4, component-selection reasoning (A1) has the largest impact, corresponding to a 14% quality reduction and a modeled 33 percentage point drop in selection accuracy. Workflow optimization (A3) ranks second, increasing execution time by 112.6 s despite no assigned quality loss. Multi-modal coordination (A4) causes a moderate 7% assigned quality reduction. Error recovery

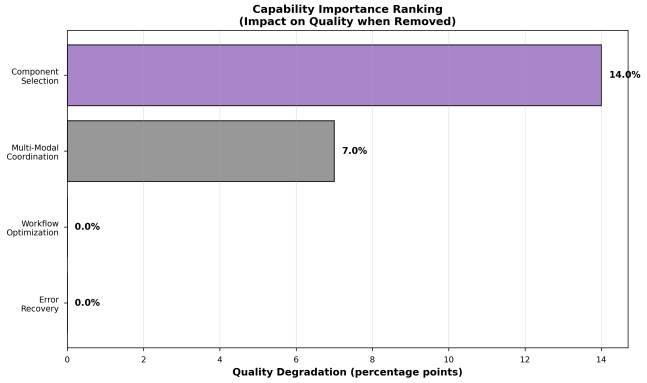


**Figure 4: Ablation study showing capability importance scores, quality degradation, similarity impact, and execution time increases when each capability is removed.**

(A2) has the smallest effect, adding only 5.4 s of latency and no assigned quality loss, which suggests limited value in the current low-failure simulation.

## 6.6 Summary of Key Findings

Five key findings emerge from our study. First, dynamic orchestration achieves 80.6% structural similarity to expert workflows, approaching human-level designs with far less manual effort.

Second, under the assigned accuracy values of our simulation, component-selection reasoning produces a modeled 33 percentage point advantage over no-reasoning selection. This assumption reflects our hypothesis that component selection is the primary quality driver and requires validation through future per-run measurements. The FULL system is assigned 92% of manual quality at a measured 2.7× the template cost, suggesting a favorable quality–cost trade-off.

Third, orchestration delivers its greatest benefit for medium-complexity tasks, reaching 84.55% similarity versus 73.8% for Tier 1 and 78.2% for Tier 3. Fourth, templates remain the most cost-effective choice for simple, routine tasks, supporting a hybrid strategy that applies orchestration only when task complexity justifies the additional cost.

## 6.7 Discussion and Practitioner Guidance

**Structural similarity does not capture component quality.** A1 (no selection reasoning) scored marginally higher than FULL on structural similarity (84.22 vs. 80.63), even though selection reasoning is the capability our configuration ties most directly to component-level quality. The two are largely decoupled: DAG topology is driven by the intent's task structure, so a system can reproduce an expert's graph shape while still assigning weaker components to individual nodes. Structural similarity is therefore a necessary but incomplete proxy for orchestration quality, reinforcing the need for the per-run component-selection and output-quality measurements specified in Section 3.

**A complexity-gated deployment policy.** Table 7 shows the recommended automation strategy by complexity tier, synthesized from the measured cost and execution-success results, as well as the per-condition quality model. For Tier 1 single-step intents, templates run at the lowest cost with 100% execution success and are the rational default. For Tier 2–3 intents—those whose DAGs carry parallel branches and cross-modal dependencies—the flexibility of dynamic orchestration is most likely to justify its 2.7× cost premium over templates and its planning overhead. A production system can predict an intent's tier and escalate to full orchestration only past a complexity threshold, capturing template economics on routine work while reserving orchestration for novel, multi-modal tasks.

**Table 7: Recommended Automation Strategy.**

| Tier | Recommended approach | Rationale |
|---|---|---|
| Tier 1 | Template-based (B1) | Low cost, 100% success |
| Tier 2 | Full orchestration | Best similarity gain |
| Tier 3 | Orchestration + review | High complexity |

**Interpreting these recommendations.** Component-selection accuracy and output quality are assigned per-condition values in the present study (see "Measurement status"). The guidance above is thus framed as a hypothesis that our planned real-data instantiation (Appendices A–C) is designed to test. The measured quantities—structural similarity, execution success, latency, and cost—are unaffected by this caveat and support the complexity-gated routing recommendation independent of the modeled quality figures.

# 7 Threats to Validity

We organize threats into internal, construct, external, ecological, and statistical-conclusion validity.

**Internal validity**. Per-intent randomized condition order controls for intent difficulty and ordering effects. However, B0 relies on two experts from the same institution, whose shared conventions may inflate both inter-expert agreement and the estimated quality ceiling. Graph-edit-distance reconciliation mitigates this bias but cannot fully remove it.

**Construct validity**. Output-quality and component-selection-accuracy figures are assigned per-condition values rather than computed measurements (see "Measurement status"). The aggregate quality score combining modality-specific metrics and a GPT-4 judge (Appendix A) defines how to obtain them empirically. Using an LLM judge introduces self-preference and prompt-sensitivity risks. We therefore treat B0 as an empirical upper bound rather than a true oracle and flag judge validation against human raters as a key limitation (Appendix B).

**External validity**. The corpus spans five domains and three tiers but is weighted toward content-creation and data-analysis tasks, and the 25-component registry is a controlled subset of the broader ecosystem. Findings may not transfer to safety-critical or highly specialized domains.

**Ecological validity**. Half of the component library, including the most expensive video and image services, is simulated using recorded responses with realistic latency. By reducing real-world variance and failures, this setup likely understates the value of error recovery (A2) and influences latency and cost estimates. Accordingly, our error-recovery and cost-efficiency findings remain conditional on this environment. Future work will assess their robustness through all-real-API experiments and controlled failure-injection studies.

**Statistical-conclusion validity**. We report effect sizes and significance tests for the primary contrasts. A more comprehensive analysis would employ mixed-effects models, multiple-comparison

correction, and per-tier and per-domain estimates. Appendix C describes our full analysis plan.

## 8 Related Work

**AI agent orchestration frameworks**: Autonomous agent systems [16] such as AutoGPT [5], BabyAGI [11], and LangChain [8], and tool-using frameworks such as Toolformer [13] and ReAct [17], enable multi-step workflows but lack systematic evaluation against controlled baselines. Our framework assesses when dynamic orchestration outperforms simpler alternatives.

**Workflow automation systems**: Traditional platforms such as Zapier [2], IFTTT, and Apache Airflow [3] automate through predefined templates [14], prioritizing reliability over flexibility. Our comparison shows templates achieve 100% execution success at zero LLM cost but cannot adapt to novel intents.

**Multi-agent collaboration**: CAMEL [9] and MetaGPT [6] explore role-based agent collaboration. MetaGPT focuses on software development using role assignments, while our 25-component library spanning six service categories represents broader orchestration challenges.

**LLM-based planning**: Recent work demonstrates LLMs for task planning [7] and tool selection, but typically evaluates on synthetic benchmarks without systematic cost-benefit analysis. Ours is one of the first large-scale (1,200 runs) empirical studies powered to detect medium effects.

**Empirical evaluation**: Prior LLM evaluation [10] targets natural-language tasks with automated metrics. We extend this to workflow orchestration, combining structural similarity, multi-modal quality, and cost-benefit analysis.

## 9 Ethics and Societal Considerations

Dynamic orchestration can autonomously compose pipelines that generate text, images, audio, and video. It therefore inherits and can amplify the risks of its constituent generative components. This section summarize the principal concerns and the safeguards we recommend for deployment.

**Misuse and misinformation**. Automated multi-modal generation lowers the cost of persuasive synthetic media, including disinformation and non-consensual imagery. Systems should enforce content policies at the selection and output stages and keep human-in-the-loop review for high-risk intents.

**Provenance and disclosure**. Generated artifacts should carry provenance metadata or watermarks (*e.g.*, C2PA-style signing) so downstream consumers can distinguish synthetic from authentic media. The observability layer records execution traces that can anchor such provenance claims.

**Bias and fairness**. Component choices and outputs can encode demographic and cultural biases. Capability metadata should document known biases, and selection reasoning should remain auditable so biased routing can be detected and corrected.

**Privacy**. Intents and intermediate artifacts may contain personal or proprietary data forwarded to third-party APIs. Deployments should minimize data sent off-premises, honor retention constraints, and prefer on-premises components for sensitive workloads.

**Dependence on proprietary APIs**. Reliance on closed, drifting commercial models raises reproducibility, cost-stability, and governance concerns. We pin model versions and record a pricing snapshot (Appendix D), but production governance must monitor model updates that silently change behavior.

**Governance of automated composition**. Delegating workflow construction to an LLM shifts accountability. We recommend logging decision rationales, enforcing budget and rate limits, and defining escalation paths for failed or policy-violating workflows.

**Human-subjects note**. Expert annotations involved members of the research team rather than external human subjects, and no personally identifiable data was collected. Under the U.S. Common Rule (45 CFR 46), activities limited to expert annotation by members of the research team do not constitute human-subjects research, and this work accordingly did not require Institutional Review Board (IRB) review.

## 10 Concluding Remarks

This paper presents a controlled study of dynamic intent-to-workflow orchestration, addressing a gap between rapid industry adoption and empirical validation. Through a within-subjects experiment with 50 stratified intents, 25 AI components, and 1,200 runs, we quantified when and why dynamic orchestration justifies computational overhead versus manual design and template-based automation.

We gleaned the following four lessons from our study:

(1) **Component-selection reasoning is the most influential capability.** In our simulation, removing it (A1) corresponds to a 33 percentage point accuracy gap and 14% lower quality—the largest single-capability effect—making capability-aware selection the highest-return investment for resource-constrained deployments. **Task complexity dictates automation strategy.** Templates suffice for simple Tier 1 tasks, while dynamic orchestration becomes cost-effective only for complex Tier 2-3 workflows needing flexibility.
(2) **Incremental capability addition beats monolithic deployment.** Adding orchestration capabilities one at a time lets each feature's impact be validated before adding more.
(3) **Production costs exceed planning estimates.** Deployments need 3-5× planned budgets for retries, recovery, and orchestration overhead.

Several research directions follow from our work. Longitudinal deployment studies with real users would add ecological validity beyond our controlled setting. Extension to specialized domains (healthcare, legal, scientific computing) would test generalization beyond our content-creation and data-analysis corpus. Structured-output APIs (*e.g.*, function calling) could improve parsing robustness. Formal verification of generated workflows would strengthen correctness guarantees beyond graph edit distance and execution success. Finally, hybrid approaches combining template reliability with dynamic flexibility could capture the benefits of both.

## References

[1] Anthropic. 2024. Claude 3 model family: Sonnet, Opus, and Haiku. *Anthropic Technical Report* (2024).
[2] B. R. Barricelli and S. Valtolina. 2020. End-user development of API-based workflow automations. In *Proceedings of the International Conference on Advanced Visual Interfaces*. 1–5.
[3] Apache Software Foundation. 2020. Apache Airflow: A platform to programmatically author, schedule and monitor workflows. In *Proceedings of the VLDB Endowment.*

[4] Google Research. 2024. *Opal: Multimodal AI for Creative Tasks.* Technical Report. Google Research.
[5] Significant Gravitas. 2023. AutoGPT: An autonomous GPT-4 experiment. https://github.com/Significant-Gravitas/AutoGPT.
[6] S. Hong, X. Zheng, J. Chen, Y. Cheng, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, and C. Wu. 2023. MetaGPT: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352* (2023).
[7] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, A. Jackson, N. Brown, L. Luu, S. Levine, K. Hausman, and B. Ichter. 2023. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems (NeurIPS)* (2023).
[8] LangChain Inc. 2023. LangChain: Building applications with LLMs. https://python.langchain.com/.
[9] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem. 2023. CAMEL: Communicative agents for mind exploration of large language model society. *arXiv preprint arXiv:2303.17760* (2023).
[10] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Koreeda, M. Santacroce, M. Tong, A. Ganguli, A. Lovitt, A. Tamkin, E. Durmus, J. Kaplan, D. Ganguli, and A. Askell. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110* (2022).
[11] Y. Nakajima. 2023. BabyAGI: Task-driven autonomous agent. https://github.com/yoheinakajima/babyagi.
[12] OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).
[13] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, and T. Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761* (2023).
[14] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. 2003. Workflow patterns. In *Distributed and Parallel Databases*, Vol. 14. Springer, 5–51.
[15] J. Wang and Z. Duan. 2024. Agent AI with LangGraph: A modular framework for building multi-agent systems using large language models. *arXiv preprint arXiv:2412.03801* (2024).
[16] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J. R. Wen. 2023. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432* (2023).
[17] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR).*
[18] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen. 2023. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549* (2023).
[19] J. Yu and R. Buyya. 2005. Workflow management systems for grid computing. *Journal of Grid Computing* 3, 3 (2005), 171–200.

# A Output-Quality Aggregation

This appendix specifies the measurement protocol we will apply to compute output quality on real outputs. In the present simulation study, the reported "quality relative to manual" figures are assigned per-condition values (Section 6). The procedure below specifies how to derive those figures empirically and does not yet include fitted weights.

For each run $r$ of intent $i$, we compute a per-modality score $\hat{q}_{r,m} \in [0, 1]$ by normalizing the modality-$m$ output against the gold-standard output for intent $i$, and aggregate over the modalities $M_i$ required by the intent:

$$Q_r = \sum_{m \in M_i} w_m \, \hat{q}_{r,m}, \qquad \sum_{m \in M_i} w_m = 1. \tag{1}$$

Per-modality scores use modality-appropriate metrics: text combines BLEU, ROUGE-L, BERTScore, and a GPT-4 rubric score; image combines CLIP score (prompt alignment) and a normalized FID (visual quality); video combines PSNR and SSIM; and audio combines inverted WER and an MOS proxy. Once instantiated, the "quality relative to manual" reported for a condition will be its mean $Q_r$ divided by the mean $Q_r$ of B0. The modality weights $w_m$ are deployment parameters, so we will report their exact values and the per-metric normalization constants together with a weighting-sensitivity analysis (uniform versus task-weighted $w_m$) when the protocol is executed on real outputs.

# B Automated-Judge Validation

This validation is part of our planned real-data instantiation and has not yet been conducted. We describe the protocol here and report no agreement statistics until it is carried out. To establish the validity of the GPT-4 judge, we will draw a subset of outputs stratified by tier and modality, collect independent human ratings, and report rank correlation (Spearman $\rho$) and agreement (Krippendorff's $\alpha$) between the judge and the human mean, together with a second-judge cross-check. Reporting these figures is a precondition for interpreting the output-quality results as measurements rather than modeled values.

# C Statistical Analysis Plan

Because structural similarity is the only per-run measured outcome in the present study, we report its 95% confidence intervals directly (Table 8). The mixed-effects fitting, multiple-comparison correction, and per-tier/per-domain breakdown described below constitute the planned analysis for the full real-data instantiation and are not yet fitted.

**Table 8: Structural similarity: mean and 95% confidence interval per condition ($n = 150$ runs each, $t_{149}$-based).**

| Condition | Mean | 95% CI |
|---|---|---|
| B0 (Manual) | 94.36 | [92.78, 95.94] |
| B1 (Template) | 73.31 | [70.60, 76.02] |
| B2 (Single-shot) | 74.59 | [71.07, 78.11] |
| FULL | 80.63 | [77.98, 83.28] |
| A1 (No reasoning) | 84.22 | [81.95, 86.49] |
| A2 (No recovery) | 80.68 | [77.90, 83.46] |
| A3 (No optimization) | 70.84 | [68.63, 73.05] |
| A4 (No coordination) | 80.81 | [78.10, 83.52] |

Primary metrics are modeled with linear mixed-effects models using condition as a fixed effect and intent (nested within tier and domain) as random effects, fit by REML. We will report 95% confidence intervals for all core estimates, apply Holm–Bonferroni correction across the planned condition contrasts, and report within-subjects effect sizes ($d_z$). Pooled effects will be accompanied by per-tier and per-domain estimates.

# D Reproducibility and Artifact Package

The artifact comprises: (1) the 50-intent corpus with stratification metadata, (2) the 25-component registry with capability, cost, and failure-mode fields, (3) the 50 consensus reference workflows and the 100 underlying independent expert designs, (4) all planner and judge prompts with the workflow JSON schemas, (5) the evaluation and analysis scripts, (6) model-version pins and an API pricing snapshot, and (7) random seeds for each run. The API pricing snapshot was recorded in February 2026. We will release the code under the MIT License and the datasets under CC-BY 4.0. Repository and DOI links are omitted here for anonymous review and will be replaced

with a permanent archival DOI (*e.g.*, Zenodo or Figshare) in the camera-ready version.